

**Курс «Алгоритмы и алгоритмические языки»  
1 семестр 2014/2015**

**Лекция 4**

## **Нормальные алгоритмы Маркова**

### **Определение нормального алгоритма Маркова (НАМ)**

$V$  – алфавит основных символов

$V'$  – алфавит маркеров

$\sigma, \sigma' \in (V \cup V')^*$

**Подстановка:**  $\sigma \rightarrow \sigma'$  переводит

слово  $\tau = \alpha\sigma\beta \in (V \cup V')^*$  в слово  $\tau' = \alpha\sigma'\beta \in (V \cup V')^*$

подслова  $\alpha$  и  $\beta$  могут быть пустыми ( $\varepsilon$ )

Помимо символов алфавита  $V \cup V'$  в подстановках используются метасимволы « $\rightarrow$ » (**стрелка**) отделяет левую часть подстановки от правой и « $\cdot$ » (**точка**) отмечает терминальную подстановку

## **Нормальные алгоритмы Маркова**

**Определение.** *Нормальный алгоритм Маркова (НАМ)* задается конечной последовательностью подстановок  $\{p_1, p_2, \dots, p_n\}$ .

При этом:

- (1) если применимо несколько подстановок, применяется подстановка, которая встречается в описании алгоритма раньше других;
- (2) если подстановка применима к нескольким подсловам обрабатываемого слова, выбирается самое левое подслово;
- (3) после применения терминальной подстановки алгоритм завершается;
- (4) если ни одна из подстановок неприменима, алгоритм завершается.

## Нормальные алгоритмы Маркова

**Пример НАМ.** Шифр Юлия Цезаря

$$V = \{a, b, c, \dots, z\}, V' = \{*\}.$$

$j$ -ая буква латинского алфавита шифруется  $j+h \pmod{26}$ -ой буквой того же алфавита.

Например, для  $h = 3$  подстановки имеют вид

(маркер \* помечает текущую шифруемую букву, цифра в скобках – номер правила подстановки):

$$\begin{aligned} (1) \ *A \rightarrow D^*, (2) \ *B \rightarrow E^*, (3) \ *C \rightarrow F^*, \dots, \\ (23) \ *W \rightarrow Z^*, (24) \ *X \rightarrow A^*, (25) \ *Y \rightarrow B^*, \\ (26) \ *Z \rightarrow C^*, (27) \ * \rightarrow ., (28) \ \rightarrow * \end{aligned}$$

Применим построенный НАМ к слову **CAESAR**:

$$\begin{aligned} CAESAR \xrightarrow{(28)} *CAESAR \xrightarrow{(3)} F*AESAR \xrightarrow{(1)} FD*ESAR \xrightarrow{(5)} \\ FDH*SAR \xrightarrow{(13)} FDHV*AR \xrightarrow{(1)} FDHVD*R \xrightarrow{(12)} \\ FDHVDU* \xrightarrow{(27)} FDHVDU \end{aligned}$$

## Нормальные алгоритмы Маркова

### Процедура интерпретации НАМ

- (1) Положить  $i = 0$ .
- (2) Положить  $j = 1$ .
- (3) Если правило  $p_j$  применимо к  $\sigma_i$ , перейти к шагу (5).
- (4) Положить  $j = j + 1$ . Если  $j \leq n$ , то перейти к шагу (3).  
В противном случае – остановка.
- (5) Применить  $p_j$  к  $\sigma_i$  и найти  $\sigma_{i+1}$ . Положить  $i = i + 1$ .  
Если  $p_j$  – нетерминальное правило, то перейти к 2.  
В противном случае – остановка.

Говорят, что НАМ *применим* к слову  $\sigma_0$ , если в результате произойдет остановка.

Терминальное правило содержит метасимвол  $\cdot$  (точка).

Иногда вместо  $\rightarrow\cdot$  используется метасимвол  $\mapsto$

## Нормальные алгоритмы Маркова

**Пример.** Сложение чисел в единичной системе счисления:

$$V = \{ +, / \}, V' = \{ \}$$

Правила подстановок:  $\{ | + \rightarrow + |; + | \rightarrow |; | \rightarrow \cdot | \}$

Пример применения алгоритма:

$$\begin{aligned}
 & \langle\langle | | | | + | | + | | \rangle\rangle = \langle\langle | | | | \color{red}{+} | | + | | \rangle\rangle \xRightarrow{(1)} \langle\langle | | | \color{red}{+} | | | + | | \rangle\rangle \\
 & \xRightarrow{(1)} \langle\langle | \color{red}{+} | | | | + | | \rangle\rangle \xRightarrow{(1)} \langle\langle | \color{red}{+} | | | | | + | | \rangle\rangle \\
 & \xRightarrow{(1)} \langle\langle + | | | | | | \color{red}{+} | | \rangle\rangle \xRightarrow{(1)} \langle\langle + | | | | | \color{red}{+} | | \rangle\rangle \xRightarrow{(1)} \dots \\
 & \xRightarrow{(1)} \langle\langle + \color{red}{+} | | | | | | | \rangle\rangle \xRightarrow{(2)} \langle\langle \color{red}{+} | | | | | | | \rangle\rangle \xRightarrow{(2)} \langle\langle | | | | | | | \rangle\rangle \xRightarrow{(3)}
 \end{aligned}$$

Первое правило «перегоняет» плюсы налево до упора, второе правило их «стирает», третье правило «убеждается», что плюсов не осталось.

# Нормальные алгоритмы Маркова

## Заключительные замечания

- ◇ **Тезис Маркова.** Любой алгоритм в алфавите  $V$  может быть представлен нормальным алгоритмом Маркова над алфавитом  $V$ .
- ◇ Примерно так же, как и для МТ, можно доказать алгоритмическую неразрешимость проблемы останова и самоприменимости.
- ◇ Существуют различные НАМ решения одной и той же задачи. Проблема построения алгоритма, который может определить эквивалентность любых двух НАМ, алгоритмически неразрешима.
- ◇ Можно построить универсальный НАМ  $U$ , который мог бы *интерпретировать* любой нормальный алгоритм, включая самого себя.

## **Заключительные замечания**

- ◇ Можно доказать эквивалентность двух формальных систем Тьюринга и Маркова конструктивным путем: построить универсальную МТ, которая могла бы интерпретировать любой НАМ и, наоборот, построить универсальный НАМ, который интерпретирует любую МТ.
- ◇ Существуют и другие формальные описания алгоритмов: машина Поста,  $\lambda$ -исчисление, рекурсивные функции и др. Для всех таких формальных систем доказана их эквивалентность МТ.
- ◇ МТ невозможно реализовать на *конечной* машине: МТ с лентой конечных размеров не обеспечивает реализации всех алгоритмов.
- ◇ **Тезис Тьюринга – Черча** (*основная гипотеза теории алгоритмов*). Для любой интуитивно вычислимой функции существует вычисляющая её значения МТ.

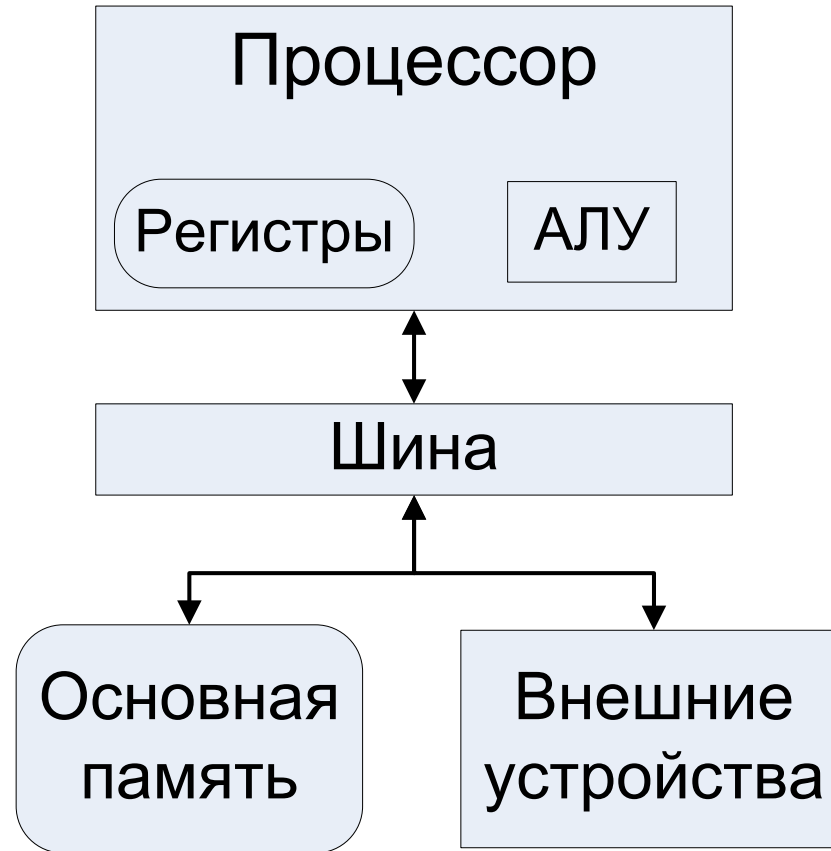


## ***Критика модели вычислений Тьюринга***

- ◆ Медленная (неускоряемая)
  - ◆ *частые копирования данных*: у нормальных МТ каждое неэлементарное действие выполняется над крайними правыми словами ленты
  - ◆ отказ от нормальных вычислений приведет к постоянному поиску данных и усложнит алгоритм
  - ◆ число состояний МТ часто зависит от числа символов в алфавите МТ

# ***Введение в язык программирования Си***

## ***Схема простейшего компьютера***



## *Язык программирования Си*

- ◆ Си разрабатывался как язык для реализации первой в мире универсальной операционной системы UNIX
- ◆ 1973 – первая версия Си
- ◆ 1978 – выход книги Б. Кернигана и Д. Ритчи «Язык программирования Си» (K&R C). Русский перевод вышел в 1985 году.
- ◆ 1989 – первый стандарт ANSI C (C89)
- ◆ 1999 – стандарт C99
- ◆ 2011 – стандарт C11 (ранее назывался C1X)

# *Введение в язык программирования Си*

## *Характеристики языка Си*

- ◆ Императивный язык
- ◆ Удобный синтаксис
- ◆ Позволяет естественно оперировать «машинными» понятиями
- ◆ Переносимость на уровне исходного кода
  - ◆ Конфигурируемость
- ◆ Хорошие системные библиотеки
- ◆ Хорошие оптимизирующие компиляторы

## *Первая программа на Си*

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    printf ("Hello, world\n");
```

```
    return 0;
```

```
}
```

Программа:

объявления переменных или функций

определения функций

## *Первая программа на Си*

```
#include <stdio.h>

int main (void)
{
    printf ("Hello, world\n");
    return 0;
}
```

Директивы препроцессора

Системные библиотеки

Строковые константы

Управляющие последовательности